# Worksheet – Georeferencing Edwardian photographers

**Before you start:** download OpenRefine from http://openrefine.org/

1. Go to Discovery and take a look at an object in COPY 1, e.g. COPY 1/5/180 (http://discovery.nationalarchives.gov.uk/details/r/C15146166)

2. We want more objects like this. Run a search for "copyright author" in COPY 1 using the advanced search. Set the year to 1909.[1]

3. Export these results and save them. This CSV file will be our first dataset.

4. Open Refine and import your CSV file. Check the preview looks reasonable (make sure you have selected 'Columns are separated by commas') and hit 'create project'.

5. We are interested in the addresses of photographers in the descriptions field. (Feel free to remove rows like the title field which contain extraneous information. Leave the file reference).

   Hit the arrow at the top of the Description column and select 'Edit Column > Add Column based on this column'

   We now have a dialogue box in which we can run an operation on this column using the 'Google Refine Expression Language' (GREL). These operations are like the string (text) operations in a programming language like Python.

6. The Photographer is the copyright author of the work. We can use this consistent piece of text to begin to split up the description. Try:

   ```
   value.split('Copyright author of work:')[1]
   ```

   (The last bit of the expression says of the two resulting parts, 0 and 1, we want the second one in the new column).

   Hit ok.

7. This column is better. But we want a clean address of the photographer in a column for mapping. Create another new column based on the new column you just created which splits the field again so that it just contains the name and address.

8. Carry out a third split to separate the photographer's name from their address. (Maybe delete your working columns to avoid confusion.)

[1] We could use a bigger year range and generate up to 10,000 results or even do this a few times and stich output together – but this would take much longer to geocode.

This is a little tricky but something like:
`value.split(value.split('‚')[0]).join('‚')`

Will (roughly) work. You may have a rogue comma:

`slice(value, 1)`

will eliminate it.

9. You now have an address you supply to Google's geoparser. Export this file from OpenRefine and import it into Google Fusion Tables (http://fusiontables.google.com). Import might be a little slow. Give it a moment.

10. Click the arrow to Change the column type in your address column to Place.

11. Select File > Geocode and select your address column.

12. Geocoding will commence. It might take a while – maybe work on some topic modelling until your map is ready for perusal…

# Worksheet – Topic modelling the future of museums

**Before you start:** download Mallet from http://mallet.cs.umass.edu/topics.php

1. Go to https://github.com/mentionthewar/Collections-and-Heritage and download the zip file of museum future articles.

2. Create a new directory in your Mallet directory with a name like museum-files and add the unzipped text files to it.

3. Open the command line (cmd in Windows) and move to the Mallet directory by heading to the root directory, perhaps by typing `cd..` a few times and the by typing something like `cd Mallet`

4. Now we can turn our files into a data file for Mallet. Run the command:

```
bin\mallet import-dir --input museum-files --output museum-future.mallet --keep-sequence --remove-stopwords
```

We now have a file we can work with.

5. Create your first topic model by entering:

bin\mallet train-topics  --input tutorial.mallet --num-topics 20 --output-state topic-state.gz --output-topic-keys tutorial_keys.txt --output-doc-topics tutorial_compostion.txt

```
bin\mallet train-topics --input museum-future.mallet --num-topics 20
```

The program will crunch a few versions of its topics before settling on a final list of 'top' words whose proximity defines the topic.

6. Try increasing the number of topics and reducing them.

   You can also add a further flag to the command:

   ```
   --num-top-words 5
   ```

   Will reduce or increase the number of top words you are shown. Does this bring the topics more into focus?

7. You will have noticed the word 'museum' occurs an awful lot in the topics. We told Mallet to remove a list of common words (stop words) when we created our Mallet file.

We can add things to this list. Create a new file called extra.txt and add it to your Mallet directory. The file could consist of the word `museum` or include other words.

Rerun the command at step 4 but add the flag:

```
--extra-stopwords extra.txt
```

If you now re-run a train-topics command, new words should move up to replace the (perhaps) redundant term ‘museum’.

8. So far we have only output text to the screen but Mallet will also output to a range of files.

   Try a command along the lines of:

```
bin\mallet train-topics --input museum-future.mallet --num-
topics 15 --output-state topic-state.gz --output-topic-keys
future_keys.txt --output-doc-topics future_composition.txt
```

This will create three files. The topic keys file contains a similar sort of output from what you have been seeing on the screen.

Take a look at the document topics file. This file contains topic weightings for all of the documents.

9. Visualising this file effectively is completed but this text file can be loaded into Google Fusion Tables and sorted by topic in order to see which documents are strongest in which topics. Looking at these relationships will help you understand what topics represent and how convincing you find this analysis.

10. We have broken the Museum Future articles by the gender of their authors. Model these collections. Do the resulting models differ? How?